

**A Final Report
Grant No. NAG-1-1112
March 1, 1990 - February 28, 1995**

ALGORITHMS FOR PARALLEL AND VECTOR COMPUTATIONS

Submitted to:

**National Aeronautics and Space Administration
Langley Research Administration
Hampton, VA 23681-0001**

Attention:

**Dr. Jules J. Lambiotte, Program Manager
ACD, M/S 125**

Submitted by:

**James M. Ortega
Charles Henderson Professor and Chair**

**Department of Computer Science
SCHOOL OF ENGINEERING AND APPLIED SCIENCE
UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA**

This is a final report on work performed under NASA Grant NAG-1-1112-FOP during the period March, 1990 through February, 1995. This grant supported four major projects, which we briefly describe.

Solution of Nonlinear Poisson-Type Equations

This was joint work with Brett Averick, who received a PhD in Applied Mathematics in January, 1991. The equation of interest was

$$\nabla(K\nabla u) = f \quad (1)$$

where K is a function of u so that (1) is nonlinear. In this case, the Jacobian matrix of the corresponding discrete equation

$$A(u)u = b(u)$$

is not symmetric although the skew-symmetric terms are small. We use this fact to approximate the Jacobian by $A(u)$, which is symmetric and positive definite. This gives rise to an approximate Newton method with fast linear convergence, rather than quadratic convergence. The linear systems at each stage are solved approximately by the incomplete Cholesky preconditioned conjugate gradient method with a variable convergence criterion; this allows relatively few conjugate gradient iterations until the iterates are near the solution. Problems on a $63 \times 63 \times 63$ grid (250,000 unknowns) are solved on a single processor of the CRAY-2 in 15 - 20 seconds, depending on the initial approximation. This work was published in [1].

Another approach was developed based on the formulation of (1) as

$$\nabla^2 \phi(u) = f. \quad (2)$$

If ϕ is a function such that $\phi'(u) = K(u)$, then

$$\nabla^2 \phi(u) = \nabla(\phi'(u)\nabla u) = \nabla(K(u)\nabla u),$$

and (2) is equivalent to (1). Thus, we obtain the solution of (1), in principle, by the two step process:

I. Solve the Poisson equation

$$\nabla^2 w = f. \quad (3)$$

II. Solve one-dimensional nonlinear equations

$$\phi(u_p) = w_p, \quad (4)$$

where w_p denotes the solution of (3) at a point P in the domain. The equations (4) can all be solved in parallel, and with no communication on a distributed memory machine. Provided that the domain is such that a Fast Poisson Solver can be used for (3), the method is very fast. This work was published in [2], and was jointly sponsored by NASA-Grants NAG-1-242, which supported Mr. Averick and NAG-1-1050.

Parallel Reduced System Conjugate Gradient Method

This was joint work with Lori Freitag, who was supported by a NASA Space Grant fellowship and the National Science Foundation, and received her PhD in Applied Mathematics in July 1992. The model differential equation is the three dimensional Helmholtz equation

$$\nabla(K\nabla u) + cu = f \quad (5)$$

where K is now a function only of the spatial variables. The domain is a parallelepiped and combinations of Dirichlet, Neumann and periodic boundary conditions are considered. This equation was proposed by T. Zang of the Theoretical Flow Physics Branch and is a kernel of various fluid codes at Langley. The differential equation is discretized by finite differences with variable grid spacing. Using the red/black ordering of the grid points, the discrete system to be solved is

$$Au = \begin{bmatrix} I & C^T \\ C & I \end{bmatrix} \begin{bmatrix} u_R \\ u_B \end{bmatrix} = \begin{bmatrix} b_R \\ b_B \end{bmatrix} \quad (6)$$

where the main diagonal elements of A have been scaled to unity. The corresponding Schur complement system for u_B is

$$(I - CC^T)u_B = b_B - Cb_R \quad (7)$$

and once this is solved, $u_R = b_R - C^T u_B$. Thus, the solution of (6) essentially reduces to that of (7), which is a system of only half the size. If A is positive definite, so is $I - CC^T$. Thus, the conjugate gradient method can be applied to (7), and this is the reduced system conjugate gradient (RSCG) method.

The main work in carrying out the RSCG method is a matrix-vector multiplication with $I - CC^T$ at each conjugate gradient iteration. This matrix is not formed; only multiplications by C and C^T are performed. Various data distributions for the Intel iPSC/860 were considered, in particular, using one, two and three dimensional mesh-connected arrays of processors. The optimal balance between message volume and the number of messages occurs for the two-dimensional array, and this configuration was used for subsequent experiments with the RSCG algorithm. The results for this algorithm show a megaflop rate of almost 450 on 128 processors, which corresponds to an efficiency of over 60%.

We also developed an analytical model of the algorithm, which can be used to predict performance on a larger number of processors and on hypothetical modifications of the Intel iPSC/860. For example, this model predicts an efficiency of over 60% on 2048 current processors and an efficiency of over 70% if the communication latency and transmission speed could both be halved.

We also considered various preconditioners for the reduced system. We concluded that it was not cost effective to form the reduced system explicitly so that preconditioners such as incomplete Cholesky factorization could be used. We tested two other preconditioners that do not require the explicit formation of the reduced system: damped Jacobi iteration and coarse grid deflation. We found that although both preconditioners reduced the number of iterations considerably the overall time did not decrease. Thus, suitable parallel preconditioners remain an open question. Results from this work were published in [3] and [4].

Orderings for Conjugate Gradient Preconditioners

In conjunction with Stephanie Stotland, who received a PhD in Applied Mathematics in September, 1993, we investigated different orderings of systems of linear equations arising from discretization of the Poisson-type differential equation

$$\nabla(K\nabla u) = f \tag{8}$$

in three dimensions. The red/black ordering has excellent parallel properties but seriously degrades the rate of convergence of the preconditioned conjugate gradient method when used with SSOR or Incomplete Cholesky preconditioning. The diagonal ordering maintains the rate of convergence and is good on vector machines such as the CRAY-2 but on distributed memory machines suffers from excessive communication and is not competitive with the red/black ordering. Similarly, the many-color orderings studied by Har-rar and Ortega for vector computers require extensive communication and are not competitive.

Orderings based on domain decomposition have shown more promise. Preliminary experiments in two-dimensions were performed on a number of such orderings: the usual block type decomposition, with and without separator sets, and strip orderings, with and without separator sets. All of these orderings performed quite well. However, the block orderings proved difficult to extend to a large number of blocks and to three dimensions. They also did not allow full use of the Eisenstat modification, which essentially eliminates the matrix-vector multiplication in the conjugate gradient algorithm. Hence, we concentrated on the strip orderings and since the ordering without separator sets had somewhat better parallel properties, we implemented this ordering (called the slab ordering) in three-dimensions.

The slab ordering in three-dimensions proved superior to the red/black ordering in a number of experiments on the iPSC/860s at NASA-Langley and Oak Ridge (up to 128 processors). These experimental results were supplemented by an analysis of the remainder matrices of the two orderings and also an analytic model. This work is pending publication [5].

SOR as a Preconditioner

Professor Ortega and Michael DeLong, a PhD candidate in Computer Science, have been studying the use of the SOR iteration as a highly parallel preconditioner for nonsymmetric systems of linear equations. A model problem, which has been used for experiments, is the convection-diffusion equation

$$\nabla^2 u + au_x + bu_y = f \tag{9}$$

discretized by finite differences on the unit square. Unless $a = b = 0$, this leads to a nonsymmetric system of linear equations. We have also considered two other convection-diffusion type equations used by J. Shalid and R. Tuminaro:

$$-u_{xx} + u_x + (1 + y^2)(-u_{yy} + u_y) = f(x, y) \quad (10)$$

$$-u_{xx} - u_{yy} + 100(x^2 u_x + y^2 u_y) - \beta u = f(x, y) \quad (11)$$

both also on the unit square with Dirichlet boundary conditions and discretized by centered differences.

The basic iteration we have been using for experiments is GMRES(m), where m is the number of steps before restart. (We also have some preliminary results with BiCGSTAB.) We precondition the system with k steps of the SOR iteration; thus, we have a two-parameter method SOR(k) – GMRES(m).

Some of the conclusions so far, based on experiments with a serial code running on an IBM RS/6000, are:

- As expected, use of the red/black ordering does not noticeably degrade the rate of convergence. Thus, the red/black ordering will allow a highly parallel implementation of the SOR iteration. The red/black ordering, however, does badly degrade the rate of convergence of ILU, when used as a preconditioner for GMRES.
- The use of a good value of ω in the SOR iteration cuts the time to convergence by roughly half. As opposed to the stand-alone SOR iteration, the convergence curve as a function of ω is very flat to the left of the optimum ω , leading to the possibility of estimating a good value of ω much more easily than with the SOR iteration. However, care is needed since an ω only slightly to the right of the optimum ω may lead to divergence.
- On equations (10) and (11), Shahid and Tuminaro used one step of Gauss-Seidel and no ω as a preconditioner in a comparison of several other preconditioners. Our results indicate that on these equations, use of several SOR steps and a reasonable value of ω improves the time by factors of 3 to 10. In this way, SOR could have been the best preconditioner for (11) and quite competitive for (10).

A report [6] on these results has been submitted for publication. This project is continuing. A parallel code for the IBM SP2 is currently under development.

Multigrid Methods

In addition to the above projects, Professor Ortega directed the PhD thesis by Robert Falgout entitled *Algebraic-Geometric Multigrid Methods for Poisson-Type Equations*, which was completed in 1991.

Publications Under the Grant

- [1] B. Averick and J. Ortega, "Solution of Nonlinear Poisson-type Equations," *Applied Numerical Mathematics*, 8, (1991), pp. 443-455.
- [2] B. Averick and J. Ortega, "Fast Solution of Nonlinear Poisson-type Equations," *SIAM J. Sci. Stat. Comp.*, 14, (1993), pp. 44-49.
- [3] L. Freitag and J. Ortega, "The RSCG Algorithm on Distributed Memory Architectures," *J. Num. Lin. Alg. Appl.*, to appear.
- [4] L. Freitag and J. Ortega, "Parallel Solution of the Generalized Helmholtz Equations," Scalable High Performance Computing Conference, Williamsburg, VA., April, 1992.
- [5] S. Stotland and J. Ortega, "Orderings for Parallel Conjugate Gradient Preconditioners," *SIAM J. Sci. Comp.*, under review.
- [6] M. DeLong and J. Ortega, "SOR as a Preconditioner," *SIAM J. Sci. Comp.*, submitted.

DISTRIBUTION LIST

- 1 - 3** **National Aeronautics and Space Administration
Langley Research Administration
Hampton, VA 23681-0001
(804) 864-5794**
- Attention: Dr. Jules J. Lambiotte, Program Director
ACD, M/S 125**
- 4** **Mr. Joseph S. Murray, Grants Officer
Acquisition Division, Mail Stop 126
National Aeronautics and Space Administration
Langley Research Administration
Hampton, VA 23681-0001
(804) 864-7709**
- 5 - 6*** **NASA Scientific and Technical Information Facility
P.O. Box 8757
Baltimore/Washington International Airport
Baltimore, MD 21240**
- 7 - 8** **J. M. Ortega**
- **** **Postaward Research Administration**
- 9 - 10** **H. Earnhardt, Clark Hall**
- 11** **SEAS Preaward Administration Files**
- *** **1 Reproducible Copy**
****** **Cover Letter Only**

JO#6277:ph